WEST Search History

Hide Items | Restore | Clear | Cancel

DATE: Monday, June 07, 2004

Hide?	Set Name	Query	Hit Count
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L15	L12 and ((light or heavy) near object)	7
☐	L14	L12 and l4	55
		DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L13	L12	0
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L12	L10 and operating	139
		DB=EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L11	L10	6
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L10	L9 or l8	177
☐	L9	L7 and (unlock$ near object)	48
☐	L8	L7 and (lock$ near object)	169
☐	L7	content service or content server or contention	24932
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L6	content service or content server or contention	18478
		DB=PGPB,USPT,USOC,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=ADJ	
☐	L5	content service or content server	8219
		DB=PGPB,USPT,USOC; PLUR=YES; OP=ADJ	
☐	L4	L3 or l2 or l1	10776
☐	L3	718/102-108.ccls.	2875
☐	L2	707/7-10.ccls.	5584
☐	L1	717/100-123,162-167.ccls.	2543

END OF SEARCH HISTORY

Hit List

Clear | Generate Collection | Print | Fwd Refs | Bkwd Refs | Generate OACS

Search Results - Record(s) 1 through 7 of 7 returned.

☐ 1. Document ID: US 20020099867 A1

Using default format because multiple data bases are involved.

L15: Entry 1 of 7 File: PGPB Jul 25, 2002

PGPUB-DOCUMENT-NUMBER: 20020099867
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020099867 A1

TITLE: Portable operating environment for information devices

PUBLICATION-DATE: July 25, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Wilkinson, Tim	Berkeley	CA	US	
Mehlitz, Peter	El Cerrito	CA	US	

US-CL-CURRENT: 719/315

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. Desc | Ir

☐ 2. Document ID: US 20020091800 A1

L15: Entry 2 of 7 File: PGPB Jul 11, 2002

PGPUB-DOCUMENT-NUMBER: 20020091800
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020091800 A1

TITLE: Portable operating environment for information devices

PUBLICATION-DATE: July 11, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Wilkinson, Tim	Berkeley	CA	US	
Mehlitz, Peter	El Cerrito	CA	US	
Fader, Tony	Santa Cruz	CA	US	

US-CL-CURRENT: 709/219; 709/246

ABSTRACT:

A client software program for use on an embedded computing device includes an operating system layer including an abstraction layer, a programming environment, and an application framework including a package manager. The client is configured to interface with a client support server including a package repository, for receiving package data for running a software package on the embedded computing device.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Ir

---

☐ 3. Document ID: US 20020070951 A1

L15: Entry 3 of 7 File: PGPB Jun 13, 2002

PGPUB-DOCUMENT-NUMBER: 20020070951
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020070951 A1

TITLE: Portable operating environment for information devices

PUBLICATION-DATE: June 13, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Wilkinson, Tim	Berkeley	CA	US	
Mihlitz, Peter	El Cernto	CA	US	

US-CL-CURRENT: 345/629

ABSTRACT:

A graphics rendering software program for use on an embedded computing device includes an application layer, a graphics toolkit, and a graphics driver for rendering a plurality of drawing surfaces, including a first drawing surface and a second drawing surface, on a display of the embedded computing device. The graphics driver is configured to render the first drawing surface at least partially overlapping the second drawing surface on the display. When the first drawing surface is rendered as partially overlapping the second drawing surface, a visible portion of the second drawing surface is computed as a set of rectangular clip segments.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw Desc | Ir

---

☐ 4. Document ID: US 20020057837 A1

L15: Entry 4 of 7 File: PGPB May 16, 2002

PGPUB-DOCUMENT-NUMBER: 20020057837
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020057837 A1

TITLE: Portable operating environment for information devices

PUBLICATION-DATE: May 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Wilkinson, Tim	Berkeley	CA	US	
Mehlitz, Peter	El Cemto	CA	US	

US-CL-CURRENT: 382/187

ABSTRACT:

A handwriting recognition software program for use with an embedded device includes an encoding module for encoding character strokes input into an input device as integer values and a character recognition module for recognizing characters as integer units.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. Desc | Ir

---

☐ 5. Document ID: US 20020057290 A1

L15: Entry 5 of 7 File: PGPB May 16, 2002

PGPUB-DOCUMENT-NUMBER: 20020057290
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20020057290 A1

TITLE: Portable operating environment for information devices

PUBLICATION-DATE: May 16, 2002

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Wilkinson, Tim	Berkely	CA	US	
Mehlitz, Peter	El Cerrito	CA	US	

US-CL-CURRENT: 345/744

ABSTRACT:

A client software program for use with an embedded device includes an application layer, a programming environment configured to render user interface component areas and query user interface component extensions from a shared delegation object, and an operating system layer. A drawing is preferably split up into a non-overridable system part and an overridable user part. The user interface component data and data used to factorize the way it is rendered are logically separated.

Full | Title | Citation | Front | Review | Classification | Date | Reference | Sequences | Attachments | Claims | KWIC | Draw. Desc | Ir

---

☐ 6. Document ID: US 20010014905 A1

L15: Entry 6 of 7 File: PGPB Aug 16, 2001

PGPUB-DOCUMENT-NUMBER: 20010014905
PGPUB-FILING-TYPE: new
DOCUMENT-IDENTIFIER: US 20010014905 A1

TITLE: Method and apparatus for managing a lock for an object

PUBLICATION-DATE: August 16, 2001

INVENTOR-INFORMATION:

NAME	CITY	STATE	COUNTRY	RULE-47
Onodera, Tamiya	Ageo-shi		JP	

US-CL-CURRENT: 718/102

ABSTRACT:

An innovative compound lock method is provided that does not reduce the processing speed attained along a frequent path. When no thread is locking an object (1), a value of 0 is stored both in a lock field and in a contention bit. Then, when a specific thread locks an object (light lock), the identifier of the thread is stored in the lock field (2). If any other thread attempts to acquire a lock before the thread designated by the thread identifier unlocks the object, SPECIAL is stored in the lock field (5), and the process is returned to (1). If a different thread attempts to acquire a lock before the designated thread unlocks the object, this causes a contention to occur in the light lock mode, and a contention bit is set to record it (3). Thereafter, when the lock mode is shifted to the fat lock mode, the contention bit is cleared (4), and if possible, the process is shifted from (4) to (1).

☐ 7. Document ID: US 6025846 A

L15: Entry 7 of 7 File: USPT Feb 15, 2000

US-PAT-NO: 6025846
DOCUMENT-IDENTIFIER: US 6025846 A
**** See image for Certificate of Correction ****

TITLE: Apparatus and method for displaying images of three-dimensional objects to first and second users from different viewpoints selected by the users

DATE-ISSUED: February 15, 2000

INVENTOR-INFORMATION:

NAME	CITY	STATE	ZIP CODE	COUNTRY
Chudley; Martin John	Somerset			GB

US-CL-CURRENT: 345/419

ABSTRACT:

A plurality of stations (25, 26, 27) are connected to a central serving processor (37) via the public switched telephone network (33). Each station (25, 26, 27) includes a processing device (28) and a monitor (29) for generating an interactive three-dimensional graphics environment. Similar environments are provided on each remote station and interaction between the stations is facilitated by transmitting transformation matrices, modified by one station, to all of the other stations. Each transformation matrix defines a transformation of an object or a light source into the displayable space.

30 Claims, 6 Drawing figures
Exemplary Claim Number: 1
Number of Drawing Sheets: 5

Terms	Documents
L12 and ((light or heavy) near object)	7

Terms used **contention** and **locking** and **stack** and **operating system** and **application framework**

Found **18,959** of **134,837**

Sort results by: relevance

Display results: expanded form

Save results to a Binder

Search Tips

☐ Open results in a new window

Try an Advanced Search

Try this search in The ACM Guide

Results 21 - 40 of 200 Result page: previous 1 **2** 3 4 5 6 7 8 9 10 next

Best 200 shown

Relevance scale

21 Design and implementation of a resource sharing system as an extension to a personal computer operating system

Rita C. Summers, Mostafa Ebrahimi, John M. Marberg, Uri Zernik

May 1985 **Proceedings of the 1985 ACM SIGSMALL symposium on Small systems**

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references

The software and hardware available today for personal computers provides a broad range of support for personal productivity, business applications, research, programming, and other activities. If personal computers are connected in a local area network, they can form a system whose total resources are very great compared to those of each computer. With appropriate system mechanisms, users can share these resources. We describe the design and implementation of a resource sharing ...

22 A technique for monitoring run-time dynamics of an operating system and a microprocessor executing user applications

Pramod V. Argade, David K. Charles, Craig Taylor

November 1994 **Proceedings of the sixth international conference on Architectural support for programming languages and operating systems**, Volume 29 , 28 Issue 11 , 5

Full text available: pdf(978.78 KB) Additional Information: full citation, abstract, references, index terms

In this paper, we present a non-invasive and efficient technique for simulating applications complete with their operating system interaction. The technique involves booting and initiating an application on a hardware development system, capturing the entire state of the application and the microprocessor at a well defined point in execution and then simulating the application on microprocessor simulators. Extensive statistics generated from the simulators on run-time dynamics of the applic ...

23 Operating system support for improving data locality on CC-NUMA compute servers

Ben Verghese, Scott Devine, Anoop Gupta, Mendel Rosenblum

September 1996 **Proceedings of the seventh international conference on Architectural support for programming languages and operating systems**, Volume 31 , 30 Issue 9 , 5

Full text available: pdf(1.26 MB) Additional Information: full citation, abstract, references, citings, index terms

The dominant architecture for the next generation of shared-memory multiprocessors is CC-NUMA (cache-coherent non-uniform memory architecture). These machines are attractive as

compute servers because they provide transparent access to local and remote memory. However, the access latency to remote memory is 3 to 5 times the latency to local memory. CC-NOW machines provide the benefits of cache coherence to networks of workstations, at the cost of even higher remote access latency. Given the larg ...

24 Elimination trees and the construction of pools and stacks: preliminary version

Nir Shavit, Dan Touitou

July 1995 **Proceedings of the seventh annual ACM symposium on Parallel algorithms and architectures**

Full text available: pdf(940.57 KB) Additional Information: full citation, references, citings, index terms

25 Low contention semaphores and ready lists

Peter J. Denning, T. Don Dennis, Jeffrey A. Brumfield

October 1981 **Communications of the ACM**, Volume 24 Issue 10

Full text available: pdf(1.17 MB) Additional Information: full citation, abstract, references, citings, index terms

A method for reducing semaphore and ready-list contention in multiprocessor operating systems is described. Its correctness is established. Its performance is compared with conventional implementations. A ready list implemented as a ring network is proposed and evaluated.

Keywords: memory contention, multiprocessing, process management, ready lists, ring networks, semaphores

26 Third Generation Computer Systems

Peter J. Denning

December 1971 **ACM Computing Surveys (CSUR)**, Volume 3 Issue 4

Full text available: pdf(3.52 MB) Additional Information: full citation, abstract, references, citings, index terms

The common features of third generation operating systems are surveyed from a general view, with emphasis on the common abstractions that constitute at least the basis for a "theory" of operating systems. Properties of specific systems are not discussed except where examples are useful. The technical aspects of issues and concepts are stressed, the nontechnical aspects mentioned only briefly. A perfunctory knowledge of third generation systems is presumed.

27 Operating System Structures to Support Security and Reliable Software

Theodore A. Linden

December 1976 **ACM Computing Surveys (CSUR)**, Volume 8 Issue 4

Full text available: pdf(3.49 MB) Additional Information: full citation, references, citings, index terms

28 Combining funnels: a new twist on an old tale...

Nir Shavit, Asaph Zemach

June 1998 **Proceedings of the seventeenth annual ACM symposium on Principles of distributed computing**

Full text available: pdf(1.29 MB) Additional Information: full citation, references, citings, index terms

29 Process management and exception handling in multiprocessor operating systems using object-oriented design techniques

Vincent Russo, Gary Johnston, Roy Campbell

January 1988 **ACM SIGPLAN Notices , Conference proceedings on Object-oriented programming systems, languages and applications**, Volume 23 Issue 11

Full text available: pdf(1.22 MB) Additional Information: full citation, abstract, references, citings, index terms

The programming of the interrupt handling mechanisms, process switching primitives, scheduling mechanisms, and synchronization primitives of an operating system for a multiprocessor require both efficient code in order to support the needs of high-performance or real-time applications and careful organization to facilitate maintenance. Although many advantages have been claimed for object-oriented class hierarchical languages and their corresponding design methodologies, the application of ...

30 A survey of commercial parallel processors

Edward Gehringer, Janne Abullarade, Michael H. Gulyn

September 1988 **ACM SIGARCH Computer Architecture News**, Volume 16 Issue 4

Full text available: pdf(2.96 MB) Additional Information: full citation, abstract, citings, index terms

This paper compares eight commercial parallel processors along several dimensions. The processors include four shared-bus multiprocessors (the Encore Multimax, the Sequent Balance system, the Alliant FX series, and the ELXSI System 6400) and four network multiprocessors (the BBN Butterfly, the NCUBE, the Intel iPSC/2, and the FPS T Series). The paper contrasts the computers from the standpoint of interconnection structures, memory configurations, and interprocessor communication. Also, the share ...

31 Distributed operating systems

Andrew S. Tanenbaum, Robbert Van Renesse

December 1985 **ACM Computing Surveys (CSUR)**, Volume 17 Issue 4

Full text available: pdf(5.49 MB) Additional Information: full citation, abstract, references, citings, index terms, review

Distributed operating systems have many aspects in common with centralized ones, but they also differ in certain ways. This paper is intended as an introduction to distributed operating systems, and especially to current university research about them. After a discussion of what constitutes a distributed operating system and how it is distinguished from a computer network, various key design issues are discussed. Then several examples of current research projects are examined in some detail ...

32 Remus: a security-enhanced operating system

Massimo Bernaschi, Emanuele Gabrielli, Luigi V. Mancini

February 2002 **ACM Transactions on Information and System Security (TISSEC)**, Volume 5 Issue 1

Full text available: pdf(295.33 KB) Additional Information: full citation, abstract, references, citings, index terms

We present a detailed analysis of the UNIX system calls and classify them according to their level of threat with respect to system penetration. Based on these results, an effective mechanism is proposed to control the invocation of critical, from the security viewpoint, system calls. The integration into existing UNIX operating systems is carried out by instrumenting the code of the system calls in such a way that the execution is granted only in the case where the invoking process and the valu ...

Keywords: Access control, Linux, privileged tasks, system calls interception, system penetration

33 Real-time concurrent collection on stock multiprocessors

A. W. Appel, J. R. Ellis, K. Li

June 1988 **ACM SIGPLAN Notices , Proceedings of the ACM SIGPLAN 1988 conference on Programming Language design and Implementation**, Volume 23 Issue 7

Full text available: pdf(1.21 MB) Additional Information: full citation, abstract, references, citings, index terms

We've designed and implemented a copying garbage-collection algorithm that is efficient, real-time, concurrent, runs on commercial uniprocessors and shared-memory multiprocessors, and requires no change to compilers. The algorithm uses standard virtual-memory hardware to detect references to "from space" objects and to synchronize the collector and mutator threads. We've implemented and measured a prototype running on SRC's 5-processor Firefly. It will be straightforward to merg ...

34 Fault Tolerant Operating Systems

Peter J. Denning

December 1976 **ACM Computing Surveys (CSUR)**, Volume 8 Issue 4

Full text available: pdf(2.69 MB) Additional Information: full citation, references, citings, index terms

35 Transactional lock-free execution of lock-based programs

Ravi Rajwar, James R. Goodman

October 2002 **Tenth international conference on architectural support for programming languages and operating systems on Proceedings of the 10th international conference on architectural support for programming languages and operating systems (ASPLOS-X)**, Volume 36 , 30 , 37 Issue 5 , 5 , 10

Full text available: pdf(1.61 MB) Additional Information: full citation, abstract, references, citings

This paper is motivated by the difficulty in writing correct high-performance programs. Writing shared-memory multi-threaded programs imposes a complex trade-off between programming ease and performance, largely due to subtleties in coordinating access to shared data. To ensure correctness programmers often rely on conservative locking at the expense of performance. The resulting serialization of threads is a performance bottleneck. Locks also interact poorly with thread scheduling and faults, r ...

36 Migration: Luna: a flexible Java protection system

Chris Hawblitzel, Thorsten von Eicken

December 2002 **ACM SIGOPS Operating Systems Review**, Volume 36 Issue SI

Full text available: pdf(1.39 MB) Additional Information: full citation, abstract, references

Extensible Java systems face a difficult trade-off between sharing and protection. On one hand, Java's ability to run different protection domains in a single virtual machine enables domains to share data easily and communicate without address space switches. On the other hand, unrestricted sharing blurs the boundaries between protection domains, making it difficult to terminate domains and enforce restrictions on resource usage. Existing solutions to these problems restrict sharing in an ad-hoc ...

37 Quartz: a tool for tuning parallel program performance

Thomas E. Anderson, Edward D. Lazowska

April 1990 **ACM SIGMETRICS Performance Evaluation Review , Proceedings of the 1990 ACM SIGMETRICS conference on Measurement and modeling of computer systems**, Volume 18 Issue 1

Full text available: pdf(1.51 MB) Additional Information: full citation, abstract, references, citings, index terms

Initial implementations of parallel programs typically yield disappointing performance. Tuning to improve performance is thus a significant part of the parallel programming process. The effort required to tune a parallel program, and the level of performance that eventually is achieved, both depend heavily on the quality of the instrumentation that is available to the programmer. This paper describes Quartz, a new tool for tuning parallel program performance on shared memory mult ...

38 Reflections on an operating system design

Butler W. Lampson, Howard E. Sturgis

May 1976 **Communications of the ACM**, Volume 19 Issue 5

Full text available: pdf(1.57 MB) Additional Information: full citation, abstract, references, citings

The main features of a general purpose multiaccess operating system developed for the CDC 6400 at Berkeley are presented, and its good and bad points are discussed as they appear in retrospect. Distinctive features of the design were the use of capabilities for protection, and the organization of the system into a sequence of layers, each building on the facilities provided by earlier ones and protecting itself from the malfunctions of later ones. There were serious problems in maintaining ...

Keywords: capabilities, faults, layering domains, memory hierarchy, operating system, protection

39 Diffracting trees

Nir Shavit, Asaph Zemach

November 1996 **ACM Transactions on Computer Systems (TOCS)**, Volume 14 Issue 4

Full text available: pdf(729.57 KB) Additional Information: full citation, abstract, references, citings, index terms

Shared counters are among the most basic coordination structures in multiprocessor conputation, with applications ranging from barrier synchronization to concurrent-data-structure design. This article introduces diffracting trees, novel data structures for share counting and load balancing in a distributed/parallel environment. Empirical evidence, collected on a simulated distributed shared-memory machine and several simulated message-passing architectures, shows that diffracting trees scal ...

Keywords: contention, counting networks, index distribution, lock free, wait free

40 Performance evaluation for multiprocessors programmed using monitors

B. J. Lucier

May 1988 **Proceedings of the 1988 ACM SIGMETRICS conference on Measurement and modeling of computer systems**

Full text available: pdf(666.89 KB) Additional Information: full citation, abstract, references, index terms

We present a classification of synchronization delays inherent in multiprocessor systems programmed using the monitor paradigm. This characterization is useful in relating performance of such systems to algorithmic parameters in subproblems such as domain decomposition. We apply this approach to a parallel, adaptive grid code for solving the equations of one-dimensional gas dynamics implemented on shared memory multiprocessors such as the Encore Multimax.